

Motion Deblurring for Plenoptic Images

Paramanand Chandramouli, Paolo Favaro and Daniele Perrone

Abstract—We address for the first time the issue of motion blur in light field images captured from plenoptic cameras. We propose a solution to the estimation of a sharp high resolution scene radiance given a blurry light field image, when the motion blur point spread function is unknown, i.e., the so-called blind deconvolution problem. In a plenoptic camera, the spatial sampling in each view is not only decimated but also defocused. Consequently, current blind deconvolution approaches for traditional cameras are not applicable. Due to the complexity of the imaging model, we investigate first the case of uniform (shift-invariant) blur of Lambertian objects, i.e., when objects are sufficiently far away from the camera to be approximately invariant to depth changes and their reflectance does not vary with the viewing direction. We introduce a highly parallelizable model for light field motion blur that is computationally and memory efficient. We then adapt a regularized blind deconvolution approach to our model and demonstrate its performance on both synthetic and real light field data. Our method handles practical issues in real cameras such as radial distortion correction and alignment within an energy minimization framework.

Index Terms—Plenoptic camera, light field image, motion blur, blind deconvolution.

I. INTRODUCTION

An important outcome of advances in computational imaging is that of a plenoptic camera which can directly capture the light field information of a scene. In the past few years, plenoptic cameras have entered into the realm of consumer photography [1], [2]. These camera are equipped with capabilities, such as 3D reconstruction and digital refocusing, not possible in traditional devices. This has led to an increased interest in the scientific community in high-quality light field reconstruction. As these commercial cameras are portable, camera shake is sometimes unavoidable and may result in blurry light field images. Similarly, moving objects can also cause the images to appear blurry.

Until now most research works on light field (LF) imaging have focused on depth estimation and super-resolution. There are no existent algorithms to handle motion blur in LF images. In contrast, motion blur in conventional cameras has been widely studied and current methods achieve remarkable results (see, for instance, [14], [39], [50], [28], [7]). Unfortunately, the imaging mechanism of a conventional camera and LF camera are quite different. Due to the additional microlens array between the main lens and the sensors in the LF camera an LF image consists of a rearranged set of views of the scene that are highly under-sampled and blurred [33]. Consequently, motion deblurring methods that are applicable for conventional images cannot be adapted in a straightforward manner.

In this paper, we propose a motion deblurring scheme for images captured from a microlens-array based light field camera. An LF image can be related to the high resolution scene texture through a space-variant point spread function (PSF) which models the image formation mechanism. In addition, the sharp scene texture is related to the blurry texture in terms of a motion blur PSF. Modeling the LF image generation by taking into account these effects turns out to be very computationally intensive and memory inefficient. However, we show that, when restricted to a constant depth scenario, it is possible to describe a motion blurred light field image as a linear combination of parallel convolutions (see sec. III-A). As a result, the model is extremely computationally and memory efficient. We address the scenario of uniform motion blur. However to handle realistic scenarios, within our framework, we allow for small variations in motion blur across the image. From a single motion blurred LF image, we solve for the high-resolution sharp scene radiance and the motion blur PSF in an energy minimization framework. We demonstrate the performance of our algorithm on real and synthetic images. The quality of results obtained by the proposed method is much higher when compared to outputs of techniques based on conventional deblurring. For real experiments, we use Lytro Illum camera. The images obtained from the LF camera suffer from radial distortion and misalignments. Our method accounts for these effects by adequately modifying the image formation model. Although, we have considered scenes that are significantly far or fronto-parallel planar, our method can be extended to handle scenes with depth variations. However, this would require estimation of depth map and handling of depth discontinuities which we consider to be beyond the scope of this paper.

II. RELATED WORK

There is no prior work that deals with motion blurred light field reconstruction. However, since our work relates to both light field reconstruction and motion deblurring, we provide a brief overview of some methods developed in both areas.

Single image motion deblurring. Motion deblurring involves the joint estimation of a sharp image and a blur kernel. Because of its ill-posedness, motion deblurring is typically addressed by enforcing prior information on the sharp image and on the blur kernel [14], [16], [27], [39], [50]. Popular choices of prior include Laplace distribution [28], total variation (TV) regularization [12] or L_0 regularization [52]. Other methods encourage sharp edges by using a shock filter [14], [50] or a dictionary of sharp edges [41]. For the blur function, the choices for prior include Gaussian distribution [50], or a sparsity-inducing distribution [16]. Given suitable priors on the sharp image and the blur function, motion deblurring is typically solved via a Maximum a Posteriori (MAP) estimation.

The authors are with Institut für Informatik und angewandte Mathematik, University of Bern, Switzerland. E-mail: chandra@iam.unibe.ch, paolo.favaro@iam.unibe.ch, perrone@iam.unibe.ch.

Levin et al. [27] have shown that a joint MAP estimation of the sharp image and blur function cannot lead to a correct estimate for a wide range of image priors. They show that marginalizing the sharp image and performing a MAP estimation on the blur function alone can correctly estimate the blur kernel. The marginalization of the sharp image is however computationally challenging, and therefore various approximations are used in practice [16], [28], [7]. Despite the theoretical analysis of Levin et al. [27], many methods successfully use a joint MAP estimation and achieve state of the art results [14], [39], [50], [36]. Recently, Perrone and Favaro [35] have clarified this apparent inconsistency. They confirmed the results of Levin et al. [27] and showed that particular implementation details make many methods in the literature work and that they do not in practice solve a joint MAP estimation.

The aforementioned deblurring methods assume uniform blur across the image plane. Techniques exist that address non-uniform blur due to rotational camera motion by introducing additional dimensions in the motion blur PSF [18], [22], [42], [49]. Hirsch et al. propose an efficient implementation scheme wherein non-uniform blur can be modeled through piece-wise uniform blurs while still retaining the global camera motion constraint [20]. Nonetheless, it has been observed in [26] that in realistic scenarios, the performance of algorithms that explicitly handle rotational motion are not necessarily better than that of the methods that assume uniform blur. Motion blur can also vary within an image because of parallax in 3D scenes. Techniques proposed in [40], [51] address this problem when there are two observations available and the camera motion is restricted to in-plane translations. While in [34], a deblurring scheme which addresses non-uniform blur for bilayer scenes using two blurred images is proposed, Hu et al. address the problem with a single observation [21]. Other methods attempt to solve general space-varying motion deblurring by estimating locally uniform blur and carefully interpolate them to cope with regions having poor texture [24], or by iteratively employing uniform deblurring and segmentation algorithms [25].

Light field capture The basic lenslet-based plenoptic camera was first developed by Adelson and Wang for inferring scene depth from a single snapshot [6]. The portable design by Ng et al. with the use of microlens arrays triggered the development of handheld light field cameras [33]. The main drawback of image reconstruction in this camera was its limited spatial resolution. To overcome this problem, Lumsdaine and Georgiev designed the focused plenoptic camera wherein the microlenses were focused at the image plane of the main lens, thereby enabling rendering at higher resolution for certain depths [31]. Perwaß and Wietzke proposed a further modification in the design which included microlens arrays with microlenses having different focal lengths [37]. Recently, Wei et al. proposed to introduce irregularity in the lens design to achieve better sampling of light fields [48]. Light field cameras have also been built by using masks instead of microlens arrays [45], [32]. Another approach to capture light field is by using camera arrays or a gantry-based moving camera [29], [44]. In this paper, we restrict our attention to microlens array-based light field cameras.

Calibration, depth estimation and super-resolution Since capturing light fields from plenoptic cameras suffers from undersampling in the spatial domain, various algorithms have been proposed to super-resolve scene radiance. In [31], the authors propose to render high resolution texture directly from light field data by projecting the LF image onto a fine grid of scene radiance. In [8], Bishop and Favaro model the image formation through geometric optics and propose a PSF-based model to relate the LF image and high resolution scene radiance along with the camera parameters and scene depth. They propose a two-step procedure to estimate the scene depth map using view correspondences and thence the high resolution scene radiance. Wanner and Goldlücke propose a technique to accurately estimate disparity using epipolar image representation and then super-resolve 4D light fields in both spatial and angular directions [47]. Broxton et al. propose a 3D-deconvolution method to reconstruct a high resolution 3D volume using a PSF-based model for the scenario of light field microscopy [11]. We also found out that this work suggests a fast computational scheme similar to the one proposed here (a very short explanation is given in a paragraph in sec.3.4 of [11]). However, our and this scheme were developed simultaneously. Moreover, our scheme includes the case of motion blur.

Recently, light field processing algorithms that also address practical issues in handling data from consumer cameras have been proposed. Danserau et al. propose a decoding, calibration and rectifying procedure for lenslet-based plenoptic cameras [15]. They develop a parameterized plenoptic camera model that relates pixels to rays in 3D space. Bok et al. propose another calibration scheme using line features [9]. Cho et al. [13] develop a method for rectification and decoding of light field data. They also develop a scheme for rendering texture at a higher resolution using a learning-based interpolation method. Xu et al. propose a decoding scheme which does not need a calibration image, instead the model parameters are estimated in an optimization scheme [53]. They also propose a 4D demosaicing scheme based on kernel regression. Huang and Cossairt propose a dictionary learning based color demosaicing scheme for plenoptic cameras [23]. In [54], Yu et al. propose to perform demosaicing while rendering the refocused image. Fiss et al. develop a scheme to refocus plenoptic images using depth adaptive splatting [17]. In [10], a hybrid imaging system with a conventional and a plenoptic camera is proposed to achieve super-resolution.

Techniques also exist that focus on obtaining accurate depth maps from plenoptic images. For example, in [43], Tao et al. combine both the correspondence and defocus cues to estimate scene depth. Motivated by the idea of active wavefront sampling, Heber et al. develop a scheme to arrive at a high quality depth map [19]. Sabater et al. propose to estimate disparity from mosaiced view correspondences instead of demosaiced views to achieve more accuracy [38]. Finally, Liang and Ramamoorthi provide an interesting analysis of the plenoptic camera using light field transport approach and propose a general framework to model all the components of the image formation process [30].

Contributions. The contributions of our work can be sum-

marized as: i) We introduce the problem of motion deblurring of light field images from a plenoptic camera and provide the first solution to it. ii) We propose a computationally and memory efficient imaging model for motion blurred LF images. iii) We solve a joint blind deconvolution and super resolution problem. iv) We handle radial distortion correction and alignment within our energy minimization framework.

III. IMAGING MODEL

In this section we introduce notation and the image formation model for a motion blurred light field image. The model relates the light field image captured by a plenoptic camera with the scene radiance via an explicit PSF. We describe the imaging model assuming uniform motion blur. Following the convention in [8], we consider the microlens array plane to be the domain on which the high-resolution scene radiance f is defined. The light field image l is defined on the image sensor plane. Let \mathbf{p} denote the coordinates of a point on the microlens array and \mathbf{x} denote a pixel location on the sensor plane. We have $\mathbf{p} = [p_1 \ p_2]^T$, and $\mathbf{x} = [x_1 \ x_2]^T$, where p_1, p_2, x_1, x_2 are integers. A pixel of the LF image $l(\mathbf{x})$ is related to texture elements $f(\mathbf{p})$ through a space-varying point spread function $h(\mathbf{x}, \mathbf{p})$ as

$$l(\mathbf{x}) = \sum_{\mathbf{p}} h(\mathbf{x}, \mathbf{p}) f(\mathbf{p}). \quad (1)$$

In general, the PSF h depends on the scene depth map and the parameters of the plenoptic camera. An explicit formula is available in [11], [8]. A relative motion between the camera and the scene during the exposure interval causes the light field image to be motion blurred. The observed motion blurred LF image can be written as a weighted sum of multiple light field images corresponding to shifted scene texture. The weights define the motion blur PSF h_m . Since the depth is constant, we can express the motion blurred light field l_m as

$$l_m(\mathbf{x}) = \sum_{\mathbf{p}} h(\mathbf{x}, \mathbf{p}) \sum_{\mathbf{q}} h_m(\mathbf{q}) f(\mathbf{p} - \mathbf{q}) = \sum_{\mathbf{p}} h(\mathbf{x}, \mathbf{p}) g(\mathbf{p}) \quad (2)$$

where $g(\mathbf{p}) \doteq \sum_{\mathbf{q}} h_m(\mathbf{q}) f(\mathbf{p} - \mathbf{q})$.

A. Convolution-based LF image generation

A direct implementation of eq. (1) is practically not feasible due to the memory and computation requirements. For instance, if the scene radiance and the light field image were of the order of mega pixels, the number of elements necessary to store h would be of the order of 10^{12} . Although h is sparse, performing the sum and product calculation in eq. (1) would still be computationally intensive. We address these shortcomings by exploiting the structure of LF PSFs. As a result we obtain an exact and efficient implementation of LF image generation that is highly parallelizable.

We first provide an intuitive explanation of the key idea that we exploit. Then, we present it formally. Although we consider a rectangular arrangement of microlenses, the idea holds even for hexagonal arrangement as seen in section III-B. Consider two LF images that are generated with the same

texture, but differ by a small shift along the X axis. If the shift in the texture f matches the distance between two microlenses, the two LF images will turn out to be exact copies of each other up to a shift equal to the number of pixels under a microlens. To illustrate this fact, we captured a few real light field images by shifting a poster from left to right while keeping our LF camera still. In the first row of Fig. 1, we show the same region cropped from three light field images. We can observe the shift of the edge pattern in these images as the poster is shifted. Now compare the first with the third image. They are (almost) identical up to a shift of 1 microlens. In the second row of Fig. 1, we show zoomed-in patches for better illustration. While the red and green patches are almost similar, the yellow patch has a shift corresponding to one microlens (with respect to the red patch). The blue patch corresponds to a texture with an intermediate shift. Thus, we can see that there is a periodic structure in the light field. Note that there is an artifact in the central microlens of the red patch. This is not due to the scene texture but because of a damaged sensor pixel.

To formalize this periodicity, we need to introduce some basic notation. Suppose there are $J \times J$ pixels under each microlens in the sensor plane. A pixel location \mathbf{x} can be written as $\mathbf{x} = \mathbf{k}J + \mathbf{j}$, where $\mathbf{j} = [j_1 \ j_2]$, $j_1, j_2 \in [0, \dots, J-1]$, and $\mathbf{k} = [k_1 \ k_2]$ with $k_1, k_2 \in \mathbb{Z}$. It should be noted that while all those pixels with the same \mathbf{j} correspond to a view or a sub-aperture image (like one image from a camera array), those with the same \mathbf{k} correspond to the image within a single microlens. We decompose the coordinates \mathbf{p} on the microlens plane in a similar fashion. If two microlenses are D units apart, then we write $\mathbf{p} = \mathbf{b}D + \mathbf{t}$, where $\mathbf{t} = [t_1 \ t_2]$, $t_1, t_2 \in [0, \dots, D-1]$, and $\mathbf{b} = [b_1 \ b_2]$, $b_1, b_2 \in \mathbb{Z}$. Note that D defines the resolution at which the scene radiance is defined. A larger value of D defines radiance on a finer grid while a smaller value would lead to a coarse resolution.

Now consider two points \mathbf{p} and \mathbf{q} located in different microlenses in such a way that their relative position with respect to the microlens center is the same (*i.e.*, both have the same value of \mathbf{t}). Then, the PSFs $h(\mathbf{x}, \mathbf{p})$ and $h(\mathbf{x}, \mathbf{q})$ will be shifted versions of each another. Technically, this is because the PSF of the light field image is given by the intersections of the blur discs of the main lens and the microlens array [8]. If there is a shift in the position of the point light source which exactly corresponds to the microlens diameter, then the resulting intersection pattern will be exactly identical to the original intersection pattern but with a shift because of the regularity of the microlens array. Thus, we can write

$$h(\mathbf{x}, \mathbf{p}) = h(\mathbf{x} - J\mathbf{y}, \mathbf{p} - D\mathbf{y}) \quad (3)$$

for any \mathbf{y} , where $\mathbf{y} = [y_1 \ y_2]$, $y_1, y_2 \in \mathbb{Z}$. The first observation is that we only need to store $J^2 \times D^2$ blur kernels, as the others can be obtained by using eq. (3). Secondly, the spatial spread of the PSF will be limited because of the limited extent of main lens blur disc. Thus, we have obtained a memory efficient representation of the PSF h . By replacing \mathbf{p} by $\mathbf{b}D + \mathbf{t}$ in

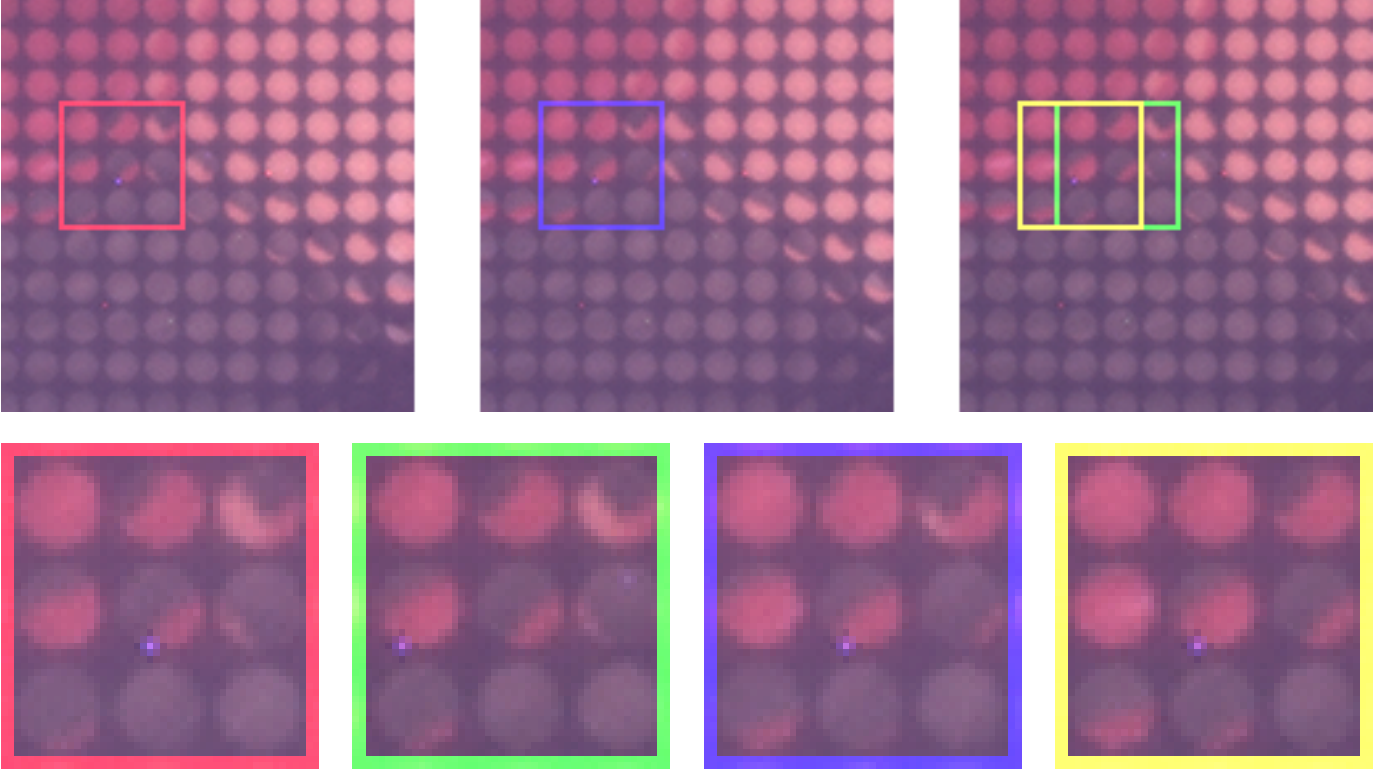


Fig. 1. Illustration of the periodicity of a LF image. Row 1: three different LF images captured by horizontally shifting texture. Zoomed-in patches whose colors indicate the portion that was cropped from the images in the first row.

eq. (2), and using eq. (3) we get

$$l_m(\mathbf{x}) = \sum_{\mathbf{t}} \sum_{\mathbf{b}} h(\mathbf{x} - \mathbf{b}J, \mathbf{t})g(\mathbf{b}D + \mathbf{t}) \quad (4)$$

Then, by expressing \mathbf{x} as $\mathbf{x} = \mathbf{k}J + \mathbf{j}$, we get

$$l_m(\mathbf{k}J + \mathbf{j}) = \sum_{\mathbf{t}} \sum_{\mathbf{b}} h(\mathbf{k}J + \mathbf{j} - \mathbf{b}J, \mathbf{t})g(\mathbf{b}D + \mathbf{t}) \quad (5)$$

Let $\hat{g}(\mathbf{b}, \mathbf{t}) \doteq g(\mathbf{b}D + \mathbf{t})$, $\hat{l}_m(\mathbf{k}, \mathbf{j}) \doteq l_m(\mathbf{k}J + \mathbf{j})$ and $\hat{h}_{\mathbf{j}}(\mathbf{k}, \mathbf{t}) \doteq h(\mathbf{k}J + \mathbf{j} - \mathbf{b}J, \mathbf{t})$ (i.e., just a rearrangement). Then, for every value of \mathbf{j} , we have

$$\hat{l}_m(\mathbf{k}, \mathbf{j}) = \sum_{\mathbf{t}} \sum_{\mathbf{b}} \hat{h}_{\mathbf{j}}((\mathbf{k} - \mathbf{b})J + \mathbf{j}, \mathbf{t}) \hat{g}(\mathbf{b}, \mathbf{t}) \quad (6)$$

and finally

$$\begin{aligned} \hat{l}_m(\mathbf{k}, \mathbf{j}) &= \sum_{\mathbf{t}} \sum_{\mathbf{b}} \hat{h}_{\mathbf{j}}(\mathbf{k} - \mathbf{b}, \mathbf{t}) \hat{g}(\mathbf{b}, \mathbf{t}) \\ &= \sum_{\mathbf{t}} \left(\hat{h}_{\mathbf{j}}(\cdot, \mathbf{t}) * \hat{g}(\cdot, \mathbf{t}) \right)(\mathbf{k}) \end{aligned} \quad (7)$$

where $*$ denotes the convolution operation. Eq. (6) indicates that we can arrive at the LF image by performing convolutions for every possible value of \mathbf{t} and \mathbf{j} . Note that these convolutions are completely independent and therefore can be executed in parallel. Also, we need only $J^2 \times D^2$ PSFs that are denoted by $\hat{h}_{\mathbf{j}}(\cdot, \mathbf{t})$ for LF image generation.

B. Hexagonal arrangement

In consumer plenoptic cameras, the microlenses are arranged on a hexagonal grid. Fig. 3 (a) shows an ideal hexagonal arrangement of microlens arrays. The basis for our convolution-based generation model was that the intersection pattern of the main lens blur disc and the microlens blur disc repeats periodically as we traverse along the microlens array plane. For the hexagonal arrangement, if we consider the set of pixels marked in red color in Fig. 2 (a) as one block and traverse in steps of this block, then the periodicity property holds. While in section III-A, for the rectangular arrangement, one block would correspond to one microlens with $J \times J$ pixels, in the hexagonal case, the size of the block is $Q \times Q'$ pixels as indicated in Fig. 2 (a). Analogously, for defining the scene texture, we consider that on the microlens array plane, there are $B \times B'$ units per block as against $D \times D$ units in the rectangular arrangement. i.e., we consider that the view index $\mathbf{j} = [j_1 \ j_2]$, $j_1 \in [0, \dots, Q' - 1]$, and $j_2 \in [0, \dots, Q - 1]$, and similarly, $\mathbf{t} = [t_1 \ t_2]$, $t_1 \in [0, \dots, B' - 1]$, and $t_2 \in [0, \dots, B - 1]$. With these changes, based on our discussion in section III-A, one can see that the LF image generation model in eq. (1) can be implemented using the convolution-based approach given in eq. (7) even for the hexagonal arrangement. In this scenario, we would need $Q \times Q' \times B \times B'$ different LF PSFs which are denoted by $\hat{h}_{\mathbf{j}}(\cdot, \mathbf{t})$.

IV. PLENOPTIC MOTION DEBLURRING

In this section, we describe our blind deconvolution algorithm wherein we solve for the sharp scene radiance from a

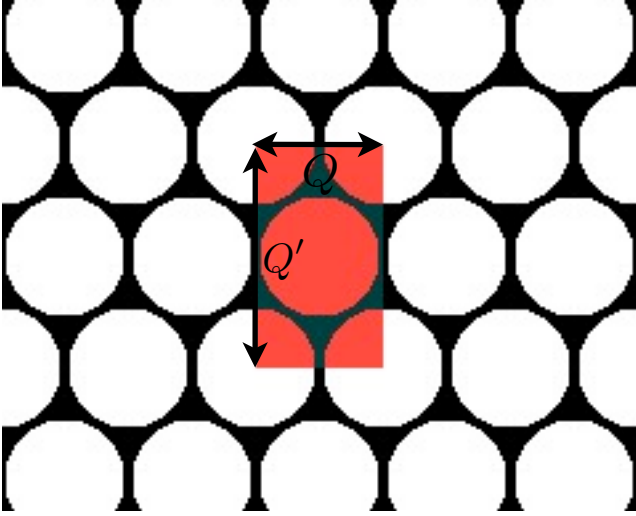


Fig. 2. Hexagonal arrangement of microlenses.

blurry light field image. Initially we assume ideal imaging conditions and uniform motion blur, and describe our technique. We subsequently discuss the modifications in our algorithm to account for small variations in motion blur and practical effects such as radial distortion, vignetting, and misalignment.

A. LF blind deconvolution

We consider that the knowledge of scene depth value and the camera parameters are available and based on these values, we calculate the LF PSF using the closed-form expression given in [8]. Then, to solve for both the motion blur PSF and the latent sharp texture from a single blurred LF image, we jointly estimate the motion blur PSF and latent texture by using an alternating minimization procedure.

Based on the model in eq. (7) of section III, the objective function can be written as

$$\min_{f, h_m} \frac{1}{2} \left\| \sum_{\mathbf{p}} h(\mathbf{x}, \mathbf{p}) \sum_{\mathbf{q}} h_m(\mathbf{q}) f(\mathbf{p} - \mathbf{q}) - l_m \right\|_2^2 + \lambda \|f\|_{BV} \quad (8)$$

subject to $h_m \succcurlyeq 0$, $\|h_m\|_1 = 1$

where $\|f\|_{BV} \doteq \int \|\nabla f(\mathbf{p})\|_2 d\mathbf{p}$, with $\nabla f \doteq [f_x \ f_y]^T$, is the total variation of f , and $\lambda > 0$ regulates the amount of total variation. The norm $\|\cdot\|_1$ corresponds to the L^1 norm.

To solve the above problem, we follow an approach similar to the projected alternating minimization algorithm of [35]. The optimization is solved via an iterative procedure based on gradient descent, where at each iteration n the current estimate of the sharp texture is given by

$$\begin{aligned} \bar{g} &\doteq h_m^{n-1} * f^{n-1} \\ \bar{l}_m(\mathbf{x}) &\doteq \sum_{\mathbf{p}} h(\mathbf{x}, \mathbf{p}) \bar{g}(\mathbf{p}) \\ \bar{g}^*(\mathbf{q}) &\doteq \sum_{\mathbf{x}} h(\mathbf{x}, \mathbf{q}) (\bar{l}_m(\mathbf{x}) - l_m(\mathbf{x})) \\ f^n &= f^{n-1} - \epsilon \left[\bar{h}_m^{n-1} * \bar{g}^* - \lambda \nabla \cdot \frac{\nabla f^{n-1}}{\|f^{n-1}\|_{BV}} \right] \end{aligned} \quad (9)$$

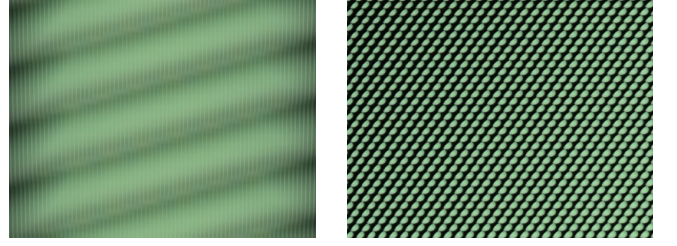


Fig. 3. Cropped white plenoptic image illustrating (a) misalignments, and (b) vignetting.

for some step $\epsilon > 0$ and where $\bar{h}_m(\mathbf{q}) = h_m(-\mathbf{q})$. For the blur kernel h_m the iteration is

$$\begin{aligned} \check{g} &\doteq h_m^{n-1} * f^n \\ \check{l}_m(\mathbf{x}) &\doteq \sum_{\mathbf{p}} h(\mathbf{x}, \mathbf{p}) \check{g}(\mathbf{p}) \\ \check{g}^*(\mathbf{q}) &\doteq \sum_{\mathbf{x}} h(\mathbf{x}, \mathbf{q}) (\check{l}_m(\mathbf{x}) - l_m(\mathbf{x})) \\ \hat{h}_m^{n-2/3} &= h_m^{n-1} - \epsilon [\check{f}^n * \check{g}^*] \end{aligned} \quad (10)$$

The last updated $\hat{h}_m^{n-2/3}$ is used to set h_m^n by using the following sequential projections

$$h_m^{n-1/3} \leftarrow \max\{\hat{h}_m^{n-2/3}, 0\}, \quad h_m^n \leftarrow \frac{h_m^{n-1/3}}{\|h_m^{n-1/3}\|_1} \quad (11)$$

Notice that all steps can be computed very efficiently via convolutions as presented in sec. III-A.

B. Practical implementation

In this subsection, we use matrix vector notation in our description. For convenience, we use the same notation f to denote a vectorized version of the sharp image, and l_m to denote the vectorized blurry light field image. The matrices H and H_m denote the light field and motion blur PSFs, respectively. The model in eq. (2) can then be re-written as $l_m = HH_m f$. However, as discussed subsequently, we modify our observation model to relate to the real world data.

In real cameras, the microlens array is not necessarily aligned with respect to the CCD sensor plane. This effect is apparent in Fig. 3 (a) which shows a crop of size 50×7728 pixels from a white image captured with Lytro Illum camera. In Fig. 3 (a), the gap between adjacent rows of microlenses is tilted instead of being horizontal. Furthermore, in real plenoptic images, the distance between adjacent microlens centers in terms of pixels need not be an integer. Hence, we need to apply an affine mapping which would align the raw image to a regular grid model as shown in Fig. 2 with both Q and Q' as integers. Let l_m denote the raw image of a blurry light field captured from a plenoptic camera. We consider that a warping matrix denoted by W when applied on LF image generated according from our model $HH_m f$ aligns it with the raw image l_m . The affine transformation remains fixed for a camera and typically its parameters are stored as metadata in plenoptic images [5].

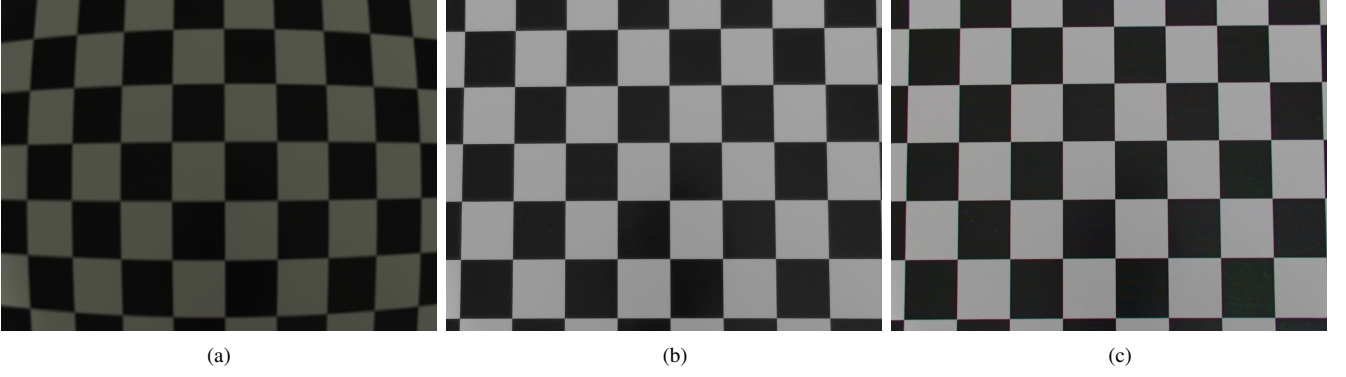


Fig. 4. Radial distortion correction. (a) Raw LF image. (b) Estimated texture with distortion correction. (c) Lytro rendering.

In microlenses that are close to the corners of the plenoptic image, we observe optical vignetting wherein a set of pixels under a microlens do not receive sufficient light from the scene. This is because light rays entering at extreme angles get blocked. Fig. 3 (b) shows an example of the vignetting effect on LF images. It contains a region cropped at the top left corner of a white plenoptic image. The image pixels under the influence of vignetting do not follow our image formation model. Hence, to avoid using the affected pixels, we use a mask M generated by thresholding a white image. The entries of the mask M will be zero at pixels that are affected by vignetting and one at other locations. To compensate for the angular sensitivity, we apply white image normalization on the captured plenoptic images. In raw images of a camera, we also find “hot pixels” due to sensor defects. The locations of the “hot pixels” can be determined by capturing a completely dark scene from the camera. These pixel locations are also set to zero in our mask M . We also mask the edge pixels of a microlens since they suffer from noise due to demosaicing [15]. To summarize, the mask M blocks the pixels affected by vignetting, “hot pixels”, and pixels present at the border of a microlens.

Yet another effect in real camera is that of radial distortion. For the LF camera which we use in our experiments, we observed significant extent of barrel distortion. Consequently, in our model, we express the LF image as

$$l_m = HRH_m f = HRg \quad (12)$$

where R is a transformation which radially distorts the motion blurred scene texture g . Since we model radial distortion to occur on the scene texture, we can use calibration techniques of conventional cameras such as [3] to determine the distortion parameters.

In our experiments, we initially determined the radial distortion parameters using the toolbox in [3]. A set of images of a checkerboard pattern was captured from different viewpoints. From each LF image, we estimated the scene texture by ignoring the radial distortion. These images were used as input to arrive at the distortion parameters. Fig. 4 shows an example wherein radial distortion correction was applied on a checkerboard plenoptic image. The raw image denoted by l is shown in Fig. 4 (a). Since motion blur is not involved, we obtain the texture by minimizing the error term $\frac{1}{2} \|M \odot (WHRf - l)\|_2^2$

along with regularization to arrive at the image shown in Fig. 4 (b). The rendering obtained by Lytro software of the same image (Fig. 4 (c)) is close to the image in Fig. 4 (b).

By considering the effect of affine warping, masking and radial distortion, the data term of our objective function can be written as

$$E_{\text{data}} = \frac{1}{2} \|M \odot (WHRH_m f - l_m)\|_2^2 \quad (13)$$

where \odot denotes the Hadamard product (point-wise multiplication). Similarly, the data cost can be expressed in terms of the motion PSF h_m by considering it to be a vector and the sharp image as a matrix denoted by F .

In order to deal with small variations in motion blur, we modify our algorithm by dividing the image into a few overlapping patches and consider that the motion blur is constant within a patch [20]. We also enforce a similarity constraint for the motion blur kernels across patches. Excluding the TV prior on the sharp texture, the cost in our formulation in terms of motion blur PSFs can then be expressed as

$$E_{\text{mod}} = \sum_{i=1}^{N_p} \frac{1}{2} \|M_p^i \odot M \odot (WHRFh_m^i - l_m)\|_2^2 + \frac{1}{2} \lambda_p \sum_{k \in \mathcal{N}_i} \|h_m^i - h_m^k\|^2 \quad (14)$$

where i corresponds to the index of a patch, N_p represents the total number of patches, the mask M_p^i takes the value of unity only in the region of the LF image corresponding to the i th patch and zero elsewhere, and \mathcal{N}_i denotes the set of indices of neighbors of the i th patch. The matrix vector product Fh_m^i can also be replaced by $H_m^i f$ when the cost is expressed in terms of f .

The gradient of the modified cost with respect to sharp scene radiance can be written as

$$\frac{\partial E_{\text{mod}}}{\partial f} = \sum_{i=1}^{N_p} H_m^i{}^T R^T H^T W^T \left(M_p^i \odot M \odot (WHRH_m^i f - l_m) \right) \quad (15)$$

The gradient with respect to i th motion PSF is given by

$$\begin{aligned} \frac{\partial E_{\text{mod}}}{\partial h_m^i} = & F^T R^T H^T W^T \left(M_p^i \odot M \odot (W H R F h_m - l_m) \right) \\ & + \lambda_p \sum_{k \in \mathcal{N}_i} (h_m^i - h_m^k) \end{aligned} \quad (16)$$

In our modified algorithm, we use the gradients evaluated according to expressions in equations (15) and (16) while implementing the update steps given in (9) and (10). Furthermore, we use a pyramid-based scheme which facilitates faster convergence.

V. EXPERIMENTAL RESULTS

We evaluate our light field blind deconvolution algorithm on several synthetic and real experiments. For synthetic experiments, we assume ideal imaging conditions as in section IV-A. We artificially generate motion blurred light field image by assuming scene texture, depth, camera parameters, and motion blur kernel. We perform real experiments with motion blurred observation captured from a handheld Lytro Illum camera. In both synthetic as well as real experiments, we apply our alternate minimization scheme to solve for the sharp texture and motion PSF with higher TV regularization. Subsequently, with the estimated PSF, we solve for the sharp texture alone with lesser TV regularization. For the purpose of comparison, we also show results obtained by a `two-step` approach. i.e., first the texture is estimated from the LF image without compensating for motion blur. The sharp image is then estimated by following conventional blind deconvolution techniques.

A. Synthetic experiments

We performed synthetic experiments on a database consisting of 12 pairs of PSFs and scene textures. For scene textures, we used `peppers.png` image of Matlab and two images from the dataset of Levin et al. [27]. For the motion blur kernels, we chose four different PSFs from [27]. We assumed realistic camera parameters and depth values to calculate the LF PSF. The values of all the parameters involved are summarized in Table II. For each combination of motion blur and texture, we generated the LF image and added 0%, 2.5% and 5% zero mean white Gaussian noise. We compared the results of our method to the output of `two-step` approach wherein the algorithm in [35] was used for blind deconvolution. The value of TV regularization parameter used was the same for both the approaches. For the noise-free experiments we set the regularization weight as $\lambda = 1.6 \cdot 10^{-3}$ for the alternating minimization scheme, and as $\lambda = 7 \cdot 10^{-4}$ to perform the final deblurring step. For the experiments with noisy images we use the values $\lambda = 2 \cdot 10^{-3}$, and $\lambda = 8 \cdot 10^{-4}$ for regularization in the initial and final stages, respectively. Note that in all our experiments, the input light field images as well as the textures take values between 0 and 1. In our comparison we used the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [46] metrics. Since between the blur function and the sharp image there is a translational ambiguity,

for each image we take the maximum PSNR and SSIM among all the possible shifts between the estimate and ground truth image. In Table I we show the mean and standard deviation of PSNR and SSIM for the results of the proposed approach and the `two-step` method. For noise-free as well as noisy scenarios, the proposed method outperforms the `two-step` method for both metrics. The `two-step` method also has a larger standard deviation. In contrast, the proposed algorithm has a small standard deviation and therefore is more consistent and robust to noise.

We show representative examples of our experiment in Fig. 5. While the images in the first row corresponds to the no-noise scenario, those in the second and third row correspond to the scenarios of 2.5% and 5% noise, respectively. The true texture and motion PSF (inserted at the bottom right corner) shown in the last column were used to generate the LF image shown in the first column. The results of the `two-step` and proposed methods are shown in third and fourth columns, respectively. For visual comparison, in the second column, we show the result of estimating the texture without compensating for motion blur (i.e., the output of the first step of the `two-step` approach). Images in Fig. 5 indicate that the performance of the proposed method is quite good.

B. Real experiments

We perform real experiments using images of motion-blurred scenes captured with a Lytro Illum camera. For reading images from the Lytro Illum camera, we use the Light Field Toolbox V0.4 software [5] developed by the authors of [15]. We initially apply intensity normalization to the raw image by using a white image of LF camera. The Lytro Illum camera consists of a database of white images for different camera settings. The Light Field Toolbox V0.4 has a function to pick the white image which is closest to a given image according to the camera settings. Intensity normalization helps to correct for the angular sensitivity of the image sensors. We captured motion blurred observations of three different real scenes from a handheld camera. In addition, we also captured another observation of the scene wherein the camera was placed on a support to avoid motion blur. We use this observation for visual evaluation of our result. In all our experiments, the scene was at a depth of more than three metres, and we used the same camera settings. We observed that for these parameters, if the depth value is beyond three meters, there would not be any variation in the LF PSF. This is because of the fact that in an LF camera the maximum observable baseline is limited to

pixels per microlens	16 × 16
vertical spacing between even rows	28 pixels
main lens focal length (F)	0.0095
pixel size	1.4 μm
main lens F-number	2.049
microlens spacing	20 μm
main lens to microlens array distance	9.8 mm
microlens array to sensor distance	47.8 μm
microlens focal length	48.0 μm

TABLE II
CAMERA PARAMETERS.



Fig. 5. Example images from the synthetic experiment. First column: LF images, second column: texture without deconvolution, third column: two-step results, fourth column: results of the proposed algorithm, and fifth column: true PSF and texture.

	Noise-free				2.5% noise				5% noise			
	PSNR		SSIM		PSNR		SSIM		PSNR		SSIM	
	proposed	two-step	proposed	two-step	proposed	two-step	proposed	two-step	proposed	two-step	proposed	two-step
μ	28.51	27.13	0.873	0.821	28.27	26.43	0.864	0.797	25.31	21.98	0.676	0.521
σ	0.83	4.71	0.021	0.173	0.850	4.02	0.021	0.154	0.961	2.93	0.035	0.105

TABLE I

IN THIS TABLE WE SHOW A COMPARISON BETWEEN THE METHODS. WE SHOW THE AVERAGE (μ) AND THE STANDARD DEVIATION (σ) OF THE PSNR AND SSIM METRICS FOR ALL SYNTHETICALLY GENERATED MOTION BLURRED LIGHT FIELD IMAGES.

the lens aperture and consequently, beyond a certain depth the disparity becomes indistinguishable.

The Lytro Illum camera has about 432×540 microlenses and the raw image has about 15 pixels per microlens. For the LF image, we define a regular hexagonal grid with $Q = 16$ and $Q' = 28$ (section III-B) and calculate the affine warping matrix for this setting. The scene texture is defined on a grid with $B = 4$, and $B' = 7$ which corresponds to about one-fourth the full sensor resolution. The LF PSF was obtained for this resolution of texture. We follow the approach discussed in section IV-B to estimate the motion PSF and thereby determine the sharp scene texture. For all experiments, the entire texture is divided into six overlapping patches in the form of a 2×3 array. i.e., we solve for six motion PSFs. In all our experiments, we used $\lambda = 1.6 \cdot 10^{-3}$ for the alternating minimization scheme, and $\lambda = 4 \cdot 10^{-4}$ for the final texture estimation. The motion PSF smoothness weight λ_p was set to be 4000.

We compare the performance of the proposed method

with two-step approach, implementation of our method with uniform motion blur assumption, and by applying the conventional blind deconvolution algorithms of [49], [7], [36] on the refocused image rendered by Lytro software. In the two-step approach, we implemented a modified version of [35] wherein PSFs were estimated patch-wise. Similar to the proposed framework, in the two-step approach, we used six overlapping patches and imposed similarity constraints on the PSFs across patches.

We tested our algorithm on plenoptic images of four different scenes as shown in Figs. 6-9. In the first row of each of the figures, from left to right, we show raw image, refocused image rendered by Lytro, and rendering of another observation that was captured without any motion blur. In the second row, we show the output of two-step approach, result obtained by uniform motion blur assumption, and the result of the proposed scheme. We show only one motion PSF (at the bottom right inset) for the proposed result and the two-step

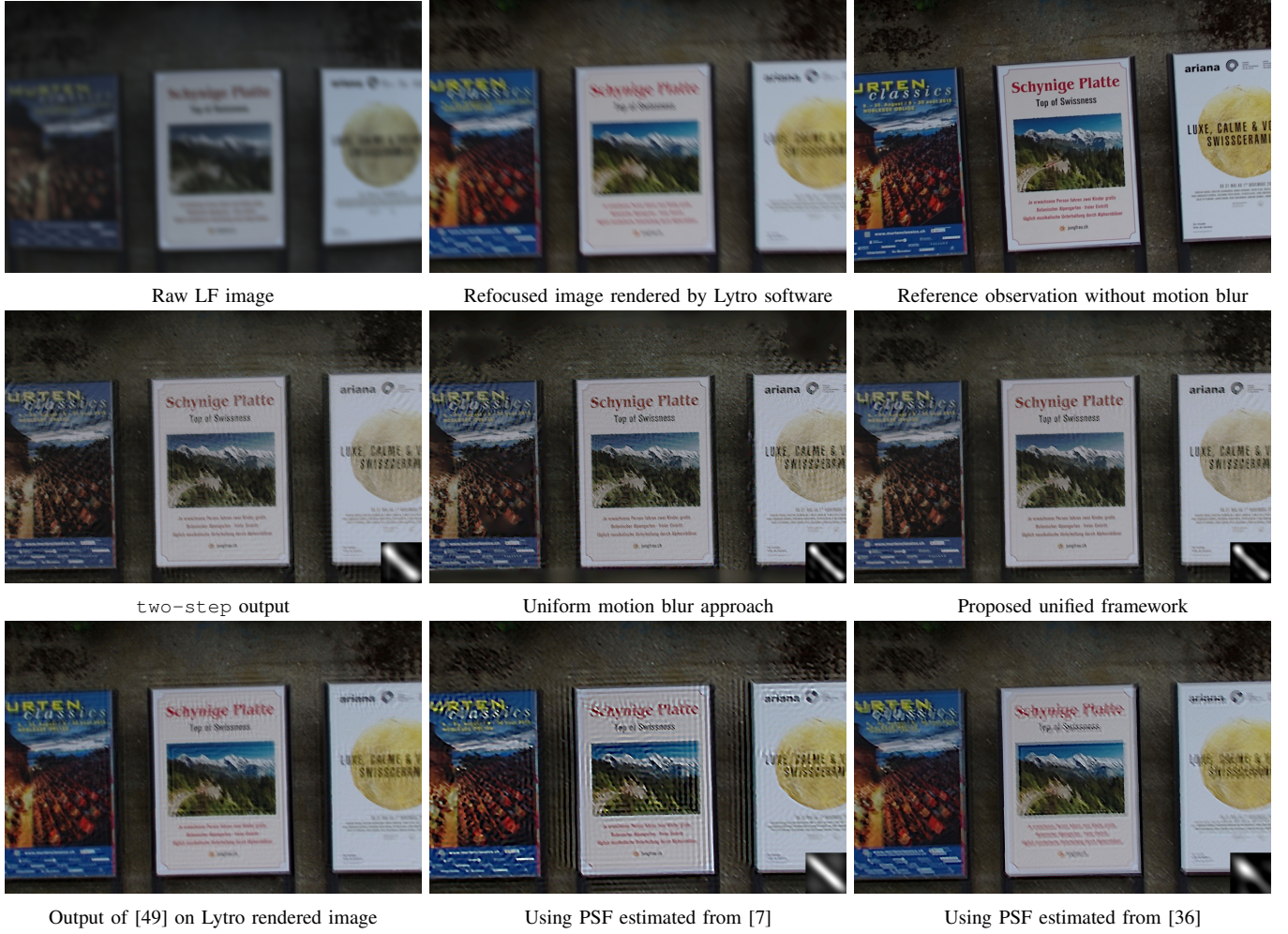


Fig. 6. Deblurring results on image of a poster with PSF shown in the insets

approach instead of six PSFs because, there were only small changes in their values and visually they looked quite similar. In the third row we first show the result of applying the non-uniform deblurring algorithm in [49] on the Lytro refocused image. The other two outputs in the third row were obtained by estimating motion blur PSFs from the Lytro rendered image using algorithms in [7] and [36], followed by TV deblurring.

In all the examples, our approach significantly improves the quality of image when compared with the Lytro rendered image. For instance, in the Poster image (Fig. 6) not only the text on the posters but also the fine texture on the background wall is similar to the texture in the unblurred observation. In the Painting image (Fig. 9), the features are sharp across the entire image in the proposed scheme as compared to other outputs. The reader is encouraged to zoom on all the observations for better examination. Although the outputs from other techniques do exhibit deblurring at instances, they also contain ringing artifacts and residual blur.

VI. CONCLUSIONS

We have presented the first motion deblurring method for plenoptic cameras. Our method extends classical blind deconvolution methods to light field cameras by modeling the

interaction of motion blur and plenoptic point spread functions. Moreover, our model is highly parallelizable and memory efficient. We follow an alternating minimization procedure to estimate the sharp and super resolved texture and the unknown motion blur. The practical issues of alignment and radial distortion are accounted for in our energy minimization framework. Experiments show that the performance of our method is quite good and outperforms approaches based on conventional blind deconvolution. Our method can be considered as a stepping stone towards more generalized scenarios of 3D scenes and non-uniform motion blur.

REFERENCES

- [1] Lytro, inc. <http://www.lytro.com/>.
- [2] Raytrix. <http://www.raytrix.de/>.
- [3] “Camera Calibration Toolbox for Matlab” http://www.vision.caltech.edu/bouguetj/calib_{_}doc/.
- [4] arXiv:1408.3686 [cs.CV]
- [5] “Light Field Toolbox v0.4” <http://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4>.
- [6] E. H. Adelson and J. Y. A. Wang, “Single lens stereo with a plenoptic camera”, *IEEE Trans. Patt. Anal. Mach. Intell.*, 14(2):99–106, 1992.
- [7] S.D. Babacan, R. Molina, M.N. Do, and A.K. Katsaggelos, “Bayesian blind deconvolution with general sparse image priors”, In *Proc. ECCV*, 2014.

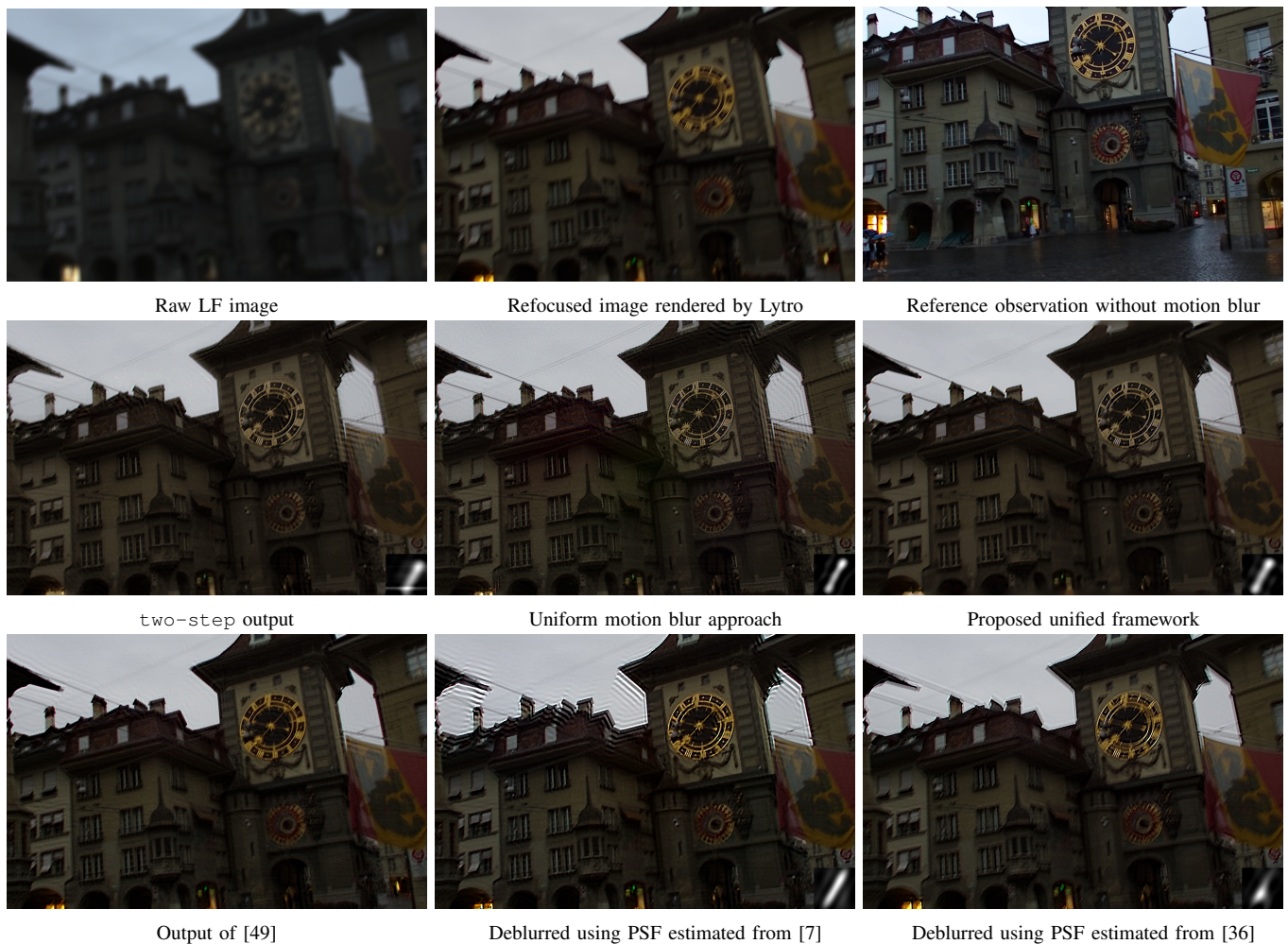


Fig. 7. Results on Zytglogge image.

- [8] T. Bishop and P. Favaro, “The light field camera: extended depth of field, aliasing and superresolution”, *IEEE Trans. Patt. Anal. Mach. Intell.*, 34 (5):972–986, 2012.
- [9] Y. Bok, H.-G. Jeon, and I. S. Kweon, “Geometric calibration of micro-lens-based light-field cameras using line features”, In *Proc. ECCV*, 2014.
- [10] V. Boominathan, K. Mitra and A. Veeraraghavan, “Improving resolution and depth-of-field of light field cameras using a hybrid imaging system”, In *Proc. ICCP*, 2014.
- [11] M. Broxton, L. Grosenick, S. Yang, N. Cohen, A. Andalman, K. Deiseroth, and M. Levoy, “Wave optics theory and 3-d deconvolution for the light field microscope”, *Opt. Express*, 21(21):25418–25439, Oct 2013.
- [12] T. F. Chan and C. K. Wong, “Total variation blind deconvolution”, *IEEE Trans. on Imag. Proc.*, vol. 7, no. 3, pp 370–375, 1998.
- [13] D. Cho, M. Kim, S. Lee, and Y. W. Tai, “Modeling the calibration pipeline of the Lytro camera for high quality light-field image reconstruction”, In *Proc. ICCV*, 2013.
- [14] S. Cho and S. Lee, “Fast motion deblurring”, *ACM Trans. Graph.*, 28 (5):1–8, 2009a. ISSN 0730-0301.
- [15] D.G. Dansereau, O. Pizarro, and S.B. Williams “Decoding, calibration and rectification for lenselet-based plenoptic cameras”, In *Proc. CVPR*, 2013.
- [16] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph”, *ACM Trans. Graph.*, 25(3):787–794, 2006a. ISSN 0730-0301.
- [17] J. Fiss, B. Curless and R. Szeliski, “Refocusing plenoptic images using depth-adaptive splatting” In *Proc. ICCP*, 2014.
- [18] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless, “Single image deblurring using motion density functions” In *Proc. ECCV*, 2010.
- [19] S. Heber, R. Ranftl, and T. Pock, “Variational shape from light field” In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 8081, pages 66–79. 2013.
- [20] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf, “Fast removal of non-uniform camera shake” In *Proc. ICCV*, 2011.
- [21] Z. Hu, L. Xu, and M. H. Yang, “Joint depth estimation and camera shake removal from single blurry image” In *Proc. CVPR*, pp. 2893–2900, 2014.
- [22] Z. Hu and M. Yang, “Fast non-uniform deblurring using constrained camera pose subspace” In *Proc. BMVC*, 2012.
- [23] X Huang and O. Cossairt, “Dictionary learning based color demosaicing for plenoptic cameras”, In *Proc. CVPRW*, 2014.
- [24] H. Ji and K. Wang, “A two-stage approach to blind spatially-varying motion deblurring”, In *Computer Vision and Pattern Recognition (CVPR)*, 2012 *IEEE Conference on*, pages 73 –80, 2012.
- [25] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee, “Dynamic scene deblurring”, In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.
- [26] R. Köhler, M. Hirsch, B. J. Mohler, B. Schölkopf, and S. Harmeling, “Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database”, In *ECCV (7)*, pages 27–40, 2012.
- [27] A Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding blind deconvolution algorithms”, *IEEE Trans. Pattern Anal. Mach. Intell.*, 33 (12):2354–2367, 2011a.
- [28] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, “Efficient marginal likelihood optimization in blind deconvolution”, In *CVPR*, pages 2657–2664, 2011b.
- [29] M. Levoy and P. Hanrahan, “Light field rendering”, In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.
- [30] C. K. Liang and R. Ramamoorthi, “A light transport framework for

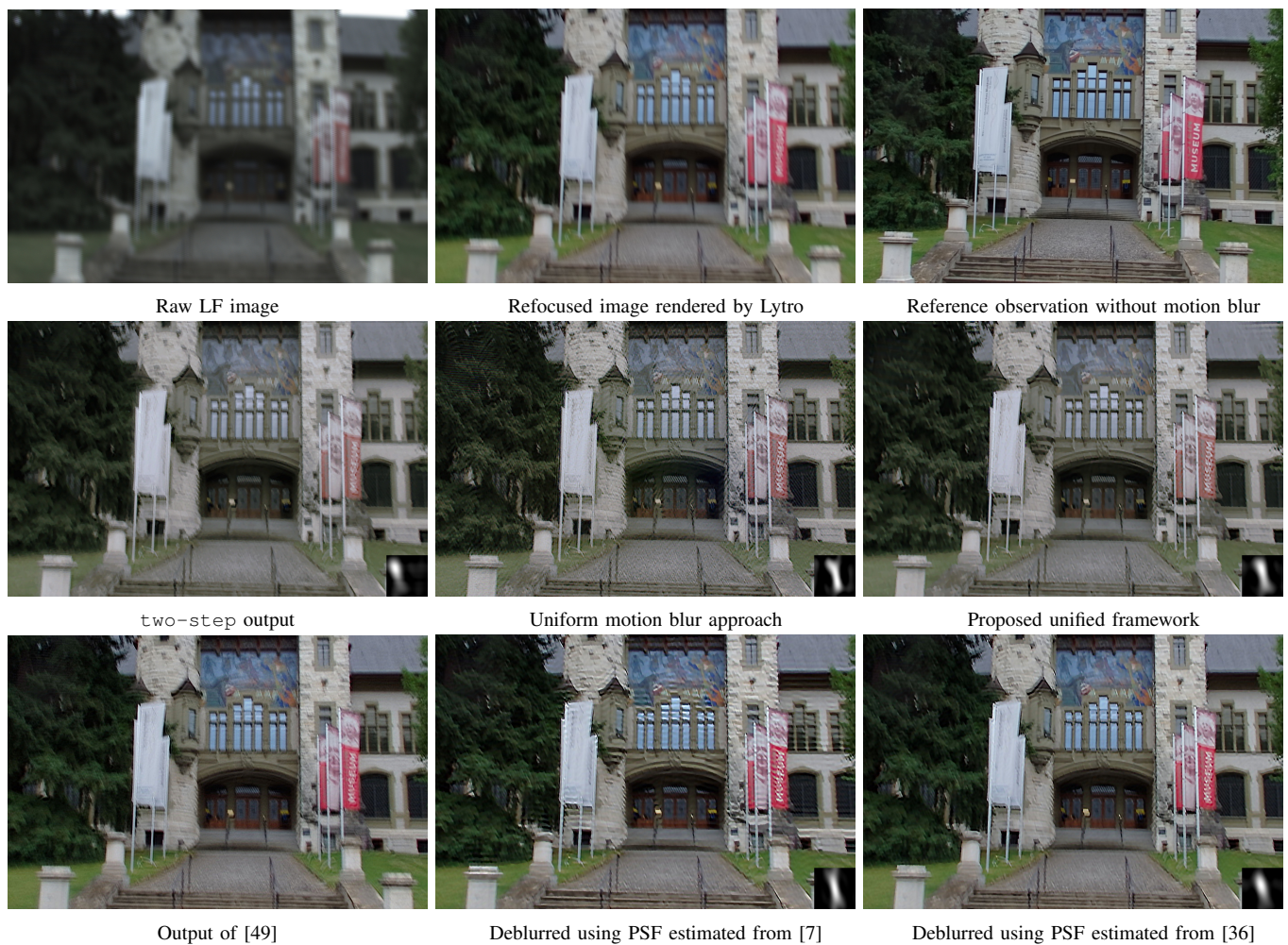


Fig. 8. Deblurring results on the image of a building.

- lenslet light field cameras” *ACM Trans. Graph.*, vol. 34, no. 2, pp. 16:1716:19, 2015.
- [31] A. Lumsdaine and T. Georgiev, “Full resolution lightfield rendering”, Technical report, Adobe Systems, 2008.
- [32] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, “Compressive light field photography using overcomplete dictionaries and optimized projections,” *ACM Trans. Graph.*, vol. 32, no. 4, 2013.
- [33] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan “Light field photography with a hand-held plenoptic camera” *Computer Science Technical Report CSTR*, 2(11), 2005.
- [34] C. Paramanand and A. N. Rajagopalan, “Non-uniform motion deblurring for bilayer scenes”, In *CVPR*, pages 1115–1122, 2013.
- [35] D. Perrone, and P. Favaro, “Total variation blind deconvolution: the devil is in the details”, In *CVPR*, pages 2909–2916, 2014.
- [36] D. Perrone, R. Diethelm and P. Favaro, “Blind deconvolution via lower-bounded logarithmic image priors”, In *EMMCVPR*, pages 112–125, 2014.
- [37] C. Perwass and L. Wietzke, “Single lens 3D-camera with extended depth-of-field”, In *Proc. SPIE*, vol. 8291, 2012.
- [38] N. Sabater, M. Seifi, V. Drazic, G. Sandri and P. Perez, “Accurate disparity estimation for plenoptic images” In *ECCVW*, 2014.
- [39] Q. Shan, J. Jia, and A. Agarwala, “High-quality motion deblurring from a single image”, *ACM Transactions on Graphics*, 27(3).
- [40] M. Sorel and J. Flusser, “Space-variant restoration of images degraded by camera motion blur”, *IEEE Trans. on Imag. Proc.*, vol. 17, no. 2, pp. 105–116, 2008.
- [41] L. Sun, S. Cho, J. Wang, and J. Hays “Edge-based blur kernel estimation using patch priors”, In *Proc. IEEE International Conference on Computational Photography*, 2013.
- [42] Y. Tai, P. Tan, and M. S. Brown, “Richardson-lucy deblurring for scenes under projective motion path”, *IEEE Trans. Patt. Anal. Mach. Intell.*, 33 (8):1603–1618, 2011.
- [43] M. W. Tao, S. Hadap, J. Malik and R. Ramamoorthi, “Depth from combining defocus and correspondence using light-field cameras”, In *Proc. ICCV*, 2013.
- [44] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, “Using plane+ parallax for calibrating dense camera arrays”, In *CVPR*, pages 1–8, 2004.
- [45] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan and J. Tumblin, “Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing,” *ACM Trans. Graph.*, vol. 26, no. 3, 2007.
- [46] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: from error visibility to structural similarity”, *IEEE Trans. on Imag. Proc.*, vol.13, pp. 600–612, 2004.
- [47] S. Wanner and B. Goldluecke, “Variational light field analysis for disparity estimation and super-resolution”, *IEEE Trans. Patt. Anal. Mach. Intell.*, 36(3):606–619, 2014.
- [48] C. K. Liang, G. Myhre, C. Pitts, K. Akeley, “Improving light field camera sample design with irregularity and aberration”, *ACM Trans. Graph.*, vol. 34, no. 4, 2015.
- [49] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, “Non-uniform deblurring for shaken images”, In *Proc. CVPR*, 2010.
- [50] L. Xu and J. Jia, “Two-phase kernel estimation for robust motion deblurring”, In *ECCV*, 2010.
- [51] L. Xu and J. Jia, “Depth-aware motion deblurring”, In *ICCP*, 2012.
- [52] L. Xu, S. Zheng, and J. Jia, “Unnatural l0 sparse representation for natural image deblurring”, In *CVPR*, 2013.
- [53] S. Xu, Z. L. Zhou and N. Devaney, “Multi-view image restoration from plenoptic raw images”, In *ACCVW*, 2014.



Fig. 9. Results on a scene consisting of a painting.

- [54] Z. Yu, J. Yu, A. Lumsdaine and T. Georgiev, "An analysis of color demosaicing in plenoptic cameras" In *CVPR*, 2012.